UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/754,010	01/08/2004	Paul Reuben day	ROC920030366US1	7130

30206 7590 07/27/2006

IBM CORPORATION
ROCHESTER IP LAW DEPT. 917
3605 HIGHWAY 52 NORTH
ROCHESTER, MN 55901-7829

EXAMINER
LOVEL, KIMBERLY M

ART UNIT	PAPER NUMBER
2167	

DATE MAILED: 07/27/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/754,010	**Applicant(s)** DAY ET AL.	
	Examiner Kimberly Lovel	**Art Unit** 2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 January 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-8,10-16 and 18-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-8,10-16 and 18-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *08 January 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date *3/29/05 5/18/05*.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. In the preliminary amendment filed on 29 March 2005, claims 1, 7, 13, 15 and 16 were amended and claims 2, 9 and 17 were cancelled.

2. Claims 1, 3-8, 10-16 and 18-20 are rejected.

Information Disclosure Statement

3. The information disclosure statements (IDS) submitted on 3/29/2005, 5/18/2005 and 3/23/2006 were filed after the mailing date of the application on 1/8/2004. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the examiner.

Claim Objections

4. Claims 2 and 8 are objected because of minor informalities.

Claim 2 is objected to because the claim has been cancelled, therefore the claim language should be deleted.

Claim 8 is objected to because is dependent on claim 9 which was cancelled.

Appropriate correction is required.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1, 3, 4, 7, 8, 10 and 12-15 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

MPEP 2106 IV.B.2.(b)

A claim that requires one or more acts to be performed defines a process. However, not all processes are statutory under 35 U.S.C. 101. Schrader, 22 F.3d at 296, 30 USPQ2d at 1460. To be statutory, a claimed computer-related process must either: (A) result in a physical transformation outside the computer for which a practical application is either disclosed in the specification or would have been known to a skilled artisan, or (B) be limited to a practical application.

Claim 1 recites a method for automatic handling of errors within a database engine, the method comprising the steps of: detecting an error while executing a query access plan; in response to detecting the error automatically rebuilding the query access plan to generate a new query access plan; and executing the new query access plan.

In the above limitation, there is no physical transformation being claimed, a practical application would be established by a useful, concrete and tangible result. For it to be a tangible result, it must be more than a thought or a computation and must have a real world value rather than being an abstract idea. The invention as recited in the claim just merely executes the query access plan and fails to have any type of output as a result of executing the plan. It is unclear as to what kind of tangible output is obtained by these limitations. Claims 5 and 6 contain the tangible result. Claims 3 and 4, which are dependent on claim 1 fail to overcome the rejection and therefore are rejected on the same grounds as claim 1.

Claim 7 recites a method for automatic handling of errors within a database engine, the method comprising the steps of: receiving an error while executing a function within a query access plan; identifying a first implementation method of the function within the query access plan; rebuilding the query access plan by replacing the first implementation method with a second implementation method and executing the new query access plan.

In the above limitation, there is no physical transformation being claimed, a practical application would be established by a useful, concrete and tangible result. For it to be a tangible result, it must be more than a thought or a computation and must have a real world value rather than being an abstract idea. The invention as recited in the claim just merely executes the query access plan and fails to have any type of output as a result of executing the plan. It is unclear as to what kind of tangible output is obtained by these limitations. Claim 11 contains the tangible result. Claims 8 and 10, which are dependent on claim 7 fail to overcome the rejection and therefore are rejected on the same grounds as claim 7.

Claim 12 recites a method for automatic handling of errors within a database engine, the method comprising the steps of: executing a query access plan comprising a plurality of functions, each function including a first implementation method; detecting a first error when executing a first function; rebuilding the query access plan to generate a new query access plan; executing the new query access plan; receiving a second error while executing the first function within the new query access plan; and rebuilding the new query access plan by replacing the first implementation method with a second implementation method of the function.

In the above limitation, there is no physical transformation being claimed, a practical application would be established by a useful, concrete and tangible result. For it to be a tangible result, it must be more than a thought or a computation and must have a real world value rather than being an abstract idea. The invention as recited in the claim just merely rebuilds the new query access plan. The new query access plan is not displayed or executed. It is unclear as to what kind of tangible output is obtained by these limitations.

Claims 13 and 15 claim a program product comprising a signal bearing medium bearing the program code. A signal is considered to be nonstatutory subject matter because it does not fall into any of the four statutory categories of invention. This is consistent with teachings of Annex IV of the "Interim Guidelines for Examination of Patent Applications for Subject Matter Eligibility" that was signed on Oct 26 and posted at http://www.uspto.gov/web/offices/pac/dapp/ogsheet.html . Claim 14, which is dependent on claims 13 fails to overcome the rejection and therefore is rejected on the same grounds as claim 13.

To expedite a complete examination of the instant application, the claims rejected under 35 U.S.C. 101 (nonstatutory) above are further rejected as set forth below in anticipation of applicant amending these claims to place them within the four statutory categories of invention.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

8. Claims 1-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over US PGPub 2002/0198867 to Lohman et al (hereafter Lohman et al) in view of the article “Efficient Mid-Query Re-Optimization of Sub-Optimal Query Execution Plans” by Kabra et al (hereafter Kabra et al).

Referring to claim 1, Lohman et al disclose a method for automatic handling of errors within a database engine (see abstract), the method comprising the step of: detecting an error while executing a query access plan (see [0018], lines 5-14). However, Lohman et al fail to explicitly disclose the further limitations of in response to detecting the error, automatically rebuilding the query access plan to generate a new query access plan and executing the new query access plan. Kabra et al disclose a similar method to that of Lohman et al for automatic handling of errors within a database engine (see abstract, lines 6-8 – the sub-optimality is considered to represent the *error*), including the further limitations of:

detecting an error while executing a query access plan (see page 109, column 2, lines 34-37 and page 110, column 1, 10-15 – the error is found during execution of the execution plan; the execution plan is considered to represent the *query access plan*);

in response to detecting the error (see page 109, column 2, line 34 – page 110, column 1, line 4 – after the error is determined the query plan is rebuilt since the remainder of the query plan is based on the estimate), automatically rebuilding the query access plan to generate a new query access plan (see page 110, column 1, lines 2-4 and lines 13-15 – upon the determination that the plan is sub-optimal, the query optimizer is re-invoked to generate a new execution plan); and

executing the new query access plan (see page 110, column 1, line 15 – the fresh new execution plan for the query is executed).

It would have been obvious to one of ordinary skill in the art to use Kabra et al's features of rebuilding the query plan after an error is found and then executing the new plan with Lohman et al's method for detecting an error in a query plan. One would have been motivated to do so in order for the query optimizer to continuously learn from any mistakes in the query plan and then have the ability to discover if the mistakes have been fixed through alterations of the query (Lohman et al: see [0017]).

Referring to claim 3, the combination of Lohman et al and Kabra et al (hereafter Lohman/Kabra) discloses the method of claim 1, wherein the error is a function check (Lohman et al: see [0077] – according to page 10, lines 3-6 of the applicant's specification, an error that halts the execution of the query is known as a function check; an error in the merge join causes a problem that has been named "implicit early out"; the error causes query to halt execution).

Referring to claim 4, Lohman/Kabra discloses the method of claim 1 further comprising the steps of:

receiving another error while executing a function within the new query access plan (Lohman et al: see [0071], lines 1-6);

identifying a first implementation method of the function within the new query access plan (Lohman et al: see [0075]); and

rebuilding the new query access plan by replacing the first implementation method with a second implementation method of the function so as to generate a rebuilt query access plan (Lohman et al: see [0075]).

Referring to claim 5, Lohman/Kabra discloses the method according to claim 1, further comprising the step of: logging information about the error, and the new query access plan (Lohman et al: see [0054] – information is logged as catalog statistics; the adjustments that will be used to repair the original catalog statistics are considered to represent the *information about the error*; storing specific adjustments with any plan skeleton is considered to represent *information about the new query access plan* since the plan skeleton represent the original plan and the adjustments represent the updated plan).

Referring to claim 6, Lohman/Kabra discloses the method according to claim 1, further comprising the step of: reporting the error (Lohman et al: see page 6, line 4 – if there is an error then the error is returned, which is considered to represent *reporting the error*).

Referring to claim 7, Lohman et al disclose a method for automatic handling of errors within a database engine (see abstract), the method comprising the steps of: receiving an error while executing a function within a query access plan (see [0018], lines 5-14); identifying a first implementation method of the function within the query access plan (Lohman et al: see [0075]), rebuilding the query access plan by replacing the first implementation method with a second implementation method of the function so as to generate a new query access plan (Lohman et al: see [0075]) and a signal bearing medium bearing the program code (see [0037], lines 3-4). However, Lohman et al fail to explicitly teach the further limitation of executing the new query access plan. Kabra et al also disclose a new query access plan including the further limitation of executing the new query access plan (see page 110, column 1, line 15 – the new execution plan for the query is executed).

It would have been obvious to one of ordinary skill in the art to use Kabra et al's feature of executing the new plan with Lohman et al's ability to rebuild a query plan. One would have been motivated to do so in order for the query optimizer to continuously learn from any mistakes in the query plan and then have the ability to discover if the mistakes have been fixed through alterations of the query (Lohman et al: see [0017]).

Referring to claim 8, Lohman/Kabra discloses the method of claim 7, wherein the function is one of a join function, an indexing function, a grouping function, and an ordering function (see [0030], lines 1-5).

Referring to claim 10, Lohman/Kabra discloses the method of claim 9, further comprising the steps of:

receiving another error while executing the function within the new query access plan (Lohman et al: see [0037] – re-executing the query to see if the new query contains errors); and

rebuilding the new query access plan by replacing the second implementation method with a third implementation method of the function (Lohman et al: see [0037]).

Referring to claim 11, Lohman/Kabra discloses the method according to claim 10 further comprising the step of: logging information about the error, the another error, and the new query access plan (Lohman et al: see [0054] – information is logged as catalog statistics; the adjustments that will be used to repair the original catalog statistics are considered to represent the *information about the error*; storing specific adjustments with any plan skeleton is considered to represent *information about the new query access plan* since the plan skeleton represent the original plan and the adjustments represent the updated plan).

Referring to claim 12, Lohman et al disclose a method for automatic handling of errors within a database engine (see abstract), the method comprising the steps of:

executing a query access plan comprising a plurality of functions (see [0037]), each function including a first implementation method; detecting a first error when executing a first function (see [0037]); rebuilding the query access plan to generate a new query access plan; receiving a second error while executing the first function within the new query access plan (see [0037]); and rebuilding the new query access plan by replacing the first implementation method with a second implementation method of the function (see [0037]). However, Lohman et al fail to explicitly teach the further limitation of executing the new query access plan. Kabra et al also disclose a new query access plan including the further limitation of executing the new query access plan (see page 110, column 1, line 15 – the new execution plan for the query is executed).

It would have been obvious to one of ordinary skill in the art to use Kabra et al's feature of executing the new plan with Lohman et al's ability to rebuild a query plan. One would have been motivated to do so in order for the query optimizer to continuously learn from any mistakes in the query plan and then have the ability to discover if the mistakes have been fixed through alterations of the query (Lohman et al: see [0017]).

Referring to claim 13, Lohman et al disclose a program product (see [0038]), comprising: a program code configured upon execution (see [0037]) to: detect an error while executing a query access plan (see [0018], lines 5-14) and a signal bearing medium bearing the program code (see [0037], lines 3-4). However, Lohman et al fail to explicitly teach the further limitation of in response to detecting the error, automatically rebuild the query access plan to generate a new query access plan.

Kabra et al also disclose detecting an error while executing a query access plan (see abstract, lines 6-8 – the sub-optimality is considered to represent the *error*) and the further limitation of in response to detecting the error (see page 109, column 2, line 34 – page 110, column 1, line 4 – after the error is determined, the query plan is rebuilt since the remainder of the query plan is based on the estimate), automatically rebuilding the query access plan to generate a new query access plan (see page 110, column 1, lines 2-4 and lines 13-15 – upon the determination that the plan is sub-optimal, the query optimizer is re-invoked to generate a new execution plan).

It would have been obvious to one of ordinary skill in the art to use Kabra et al's features of rebuilding the query plan after an error is found and then executing the new plan with Lohman et al's ability to detect an error in a query plan. One would have been motivated to do so in order for the query optimizer to continuously learn from any mistakes in the query plan and then have the ability to discover if the mistakes have been fixed through alterations of the query (Lohman et al: see [0017]).

Referring to claim 14, Lohman/Kabra discloses the program product of claim 13, wherein the program code is further configured to:

receive an error while executing a function within the new query access plan (Lohman et al: see [0071], lines 1-6);

identify a first implementation method of the function within the new query access plan (Lohman et al: see [0075]); and rebuild the new query access plan by replacing the first implementation method with a second implementation method of the function so as to generate a rebuilt query access plan (Lohman et al: see [0075]).

Referring to claim 15, Lohman et al disclose a program product (see [0038]), comprising: a program code configured upon execution (see [0037]) thereof to: receive an error while executing a function within a query access plan (see [0018], lines 5-14); identify a first implementation method of the function within the query access plan (Lohman et al: see [0075]), rebuild the query access plan by replacing the first implementation method with a second implementation method of the function so as to generate a new query access plan (Lohman et al: see [0075]) and a signal bearing medium bearing the program code (see [0037], lines 3-4). However, Lohman et al fail to explicitly teach the further limitation of executing the new query access plan. Kabra et al also disclose a new query access plan including the further limitation of executing the new query access plan (see page 110, column 1, line 15 – the new execution plan for the query is executed).

It would have been obvious to one of ordinary skill in the art to use Kabra et al's feature of executing the new plan with Lohman et al's ability to rebuild a query plan. One would have been motivated to do so in order for the query optimizer to continuously learn from any mistakes in the query plan and then have the ability to discover if the mistakes have been fixed through alterations of the query (Lohman et al: see [0017]).

Referring to claim 16, Lohman et al disclose an apparatus (see [0038], lines 1-2) comprising: at least one processor (see [0037], line 5 - computer); a memory coupled with the at least one processor (see [0037], lines 4-5); and a program code residing in memory and executed by the at least one processor (see [0037]), the program code configured to: detect an error while executing a query access plan (see [0018], lines 5-14). However, Lohman et al fail to explicitly disclose the further limitations of automatically rebuilding the query access plan, in response to detecting the error, generating a new query access plan, and executing the new query access plan. Kabra et al also disclose detecting an error while executing a query access plan (see abstract, lines 6-8 – the sub-optimality is considered to represent the *error*) and the further limitations of in response to detecting the error (see page 109, column 2, line 34 – page 110, column 1, line 4 – after the error is determined, the query plan is rebuilt since the remainder of the query plan is based on the estimate), automatically rebuilding the query access plan to generate a new query access plan (see page 110, column 1, lines 2-4 and lines 13-15 – upon the determination that the plan is sub-optimal, the query optimizer is re-invoked to generate a new execution plan).

It would have been obvious to one of ordinary skill in the art to use Kabra et al's features of rebuilding the query plan after an error is found and then executing the new plan with Lohman et al's ability to detect an error in a query plan. One would have been motivated to do so in order for the query optimizer to continuously learn from any mistakes in the query plan and then have the ability to discover if the mistakes have been fixed through alterations of the query (Lohman et al: see [0017]).

Referring to claim 18, Lohman/Kabra discloses apparatus of claim 16, wherein the error is a function check (Lohman et al: see [0077] – according to page 10, lines 3-6 of the applicant's specification, an error that halts the execution of the query is known as a function check; an error in the merge join causes a problem that has been named "implicit early out"; the error causes query to halt execution).

Referring to claim 19, Lohman/Kabra discloses the method of claim 16, wherein the program code is further configured to:

detect another error while executing a function within the new query access plan (Lohman et al: see [0037], lines 1-6);

identify a first implementation method of the function within the new query access plan (Lohman et al: see [0037]); and

rebuild the new query access plan by replacing the first implementation method with a second implementation method of the function so as to generate a rebuilt query access plan (Lohman et al: see [0037]).

Referring to claim 20, Lohman/Kabra discloses method according to claim 16, wherein the program code is further configured to: log information about the error, and the new query access plan (Lohman et al: see [0054] – information is logged as catalog statistics; the adjustments that will be used to repair the original catalog statistics are considered to represent the *information about the error*; storing specific adjustments with any plan skeleton is considered to represent *information about the new query access plan* since the plan skeleton represent the original plan and the adjustments represent the updated plan).

Referring to claim 21, Lohman/Kabra discloses the method according to claim 16, wherein the program code is further configured to: report the error (Lohman et al: see page 6, line 4 – if there is an error then the error is returned, which is considered to represent *reporting the error*).

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kimberly Lovel whose telephone number is (571) 272-2750. The examiner can normally be reached on 8:00 - 4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on (571) 272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Kimberly Lovel
Examiner
Art Unit 2167

kml
5 July 2006

Luke S Wassum
Primary Examiner
Art Unit 2167